AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (currently amended) A computing system that includes a computer-readable medium and calculates an exponent portion of a floating point data type, for use in calculating powers of two in a computer language processing environment supporting a union declaration functionality and a left shift functionality, comprising:

an input receptive of an exponent value;

a bias application module that biases the exponent value based on a selected precision of a floating point data type;

a storage module that stores the biased exponent value in a storage of the computer-readable medium ~~variable~~ having a size determined based on the selected precision;

a left shift application module that shifts the biased exponent value left by a number of bits determined based on the selected precision; and

an output module that outputs ~~returns~~ the biased exponent value from the storage ~~variable~~ as the floating point data type having the selected precision.

2. (currently amended) The system of claim 1 ~~further comprising a storage variable declaration module that declares the~~ wherein the storage ~~variable~~ can store ~~as a union of~~ an integer data type and a float data type.

3. (currently amended) The system of claim 2 wherein the selected precision is single, and ~~said~~ the storage ~~variable declaration module instantiates a union of~~ can store a thirty-two bit integer data type and a single precision floating point data type.

4. (currently amended) The system of claim 2 wherein the selected precision is double, and ~~said~~ the storage ~~variable declaration module instantiates a union of~~ can store a sixty-four bit integer data type and a double precision floating point data type.

5. (original) The system of claim 2 wherein said input is further receptive of the selected precision, said storage variable declaration module determines a bias size, a mantissa size, and the size of the storage variable based on the selected precision, said bias application module biases the exponent value based on the bias size, and said left shift application module shifts the storage variable left based on the mantissa size.

6. (original) The system of claim 1 further comprising a limit application module that limits the exponent value based on an exponent value range determined based on the selected precision.

7. (original) The system of claim 6 wherein the selected precision is single, and said limit application module determines whether the exponent value at least one of exceeds one-hundred twenty-seven and falls below negative one-hundred twenty-six, and to set the exponent value equal to at least one of one-hundred twenty-seven and negative one-hundred twenty-six when the exponent value respectively at least one of exceeds one-hundred twenty-seven and falls below negative one-hundred twenty-six.

8. (original) The system of claim 6 wherein the selected precision is double, and said limit application module determines whether the exponent value at least one of exceeds one-thousand twenty-four and falls below negative one-thousand twenty-three, and to set the exponent value equal to at least one of one-thousand twenty-four and negative one-thousand twenty-three when the exponent value respectively at least one of exceeds one-thousand twenty-four and falls below negative one-thousand twenty-three.

9. (original) The system of claim 1 wherein the selected precision is single, and said bias application module adds one-hundred twenty seven to the exponent value.

10. (original) The system of claim 1 wherein the selected precision is double, and said bias application module adds one-thousand twenty-three to the exponent value.

11. (currently amended) A computing method for calculating an exponent portion of a floating point data type, for use in calculating powers of two in a computer language processing environment supporting a union declaration functionality and a shift left functionality, comprising:

receiving an exponent value;

biasing the exponent value based on a selected precision of a floating point data type;

storing the biased exponent value in a storage ~~variable~~ of a computer-readable medium, said storage having a size determined based on the selected precision;

shifting the biased exponent value left by a number of bits determined based on the selected precision; and

~~returning~~ outputting the biased exponent value from the storage ~~variable~~ as the floating point data type having the selected precision.

12. (currently amended) The method of claim 11 ~~further comprising declaring~~ wherein the storage ~~variable as a union of~~ can store an integer data type and a float data type.

13. (currently amended) The method of claim 12 wherein the selected precision is single, and ~~said declaring includes instantiating a union of~~ the storage can store a thirty-two bit integer data type and a single precision floating point data type.

14. (currently amended) The method of claim 12 wherein the selected precision is double, and ~~said declaring includes instantiating a union of~~ the storage can store a sixty-four bit integer data type and a double precision floating point data type.

15. (original) The method of claim 11 further comprising limiting the exponent value based on an exponent value range determined based on the selected precision.

16. (original) The method of claim 15 wherein the selected precision is single, and said limiting includes:

determining whether the exponent value at least one of exceeds one-hundred twenty-seven and falls below negative one-hundred twenty-six; and

setting the exponent value equal to at least one of one-hundred twenty-seven and negative one-hundred twenty-six when the exponent value respectively at least one of exceeds one-hundred twenty-seven and falls below negative one-hundred twenty-six.

17. (original) The method of claim 15 wherein the selected precision is double, and said limiting includes:

determining whether the exponent value at least one of exceeds one-thousand twenty-four and falls below negative one-thousand twenty-three; and

setting the exponent value equal to at least one of one-thousand twenty-four and negative one-thousand twenty-three when the exponent value respectively at least one of exceeds one-thousand twenty-four and falls below negative one-thousand twenty-three.

18. (original) The method of claim 11 wherein the selected precision is single, and said biasing includes adding one-hundred twenty seven to the exponent value.

19. (original) The method of claim 11 wherein the selected precision is double, and said biasing includes adding one-thousand twenty-three to the exponent value.

20. (currently amended) The method of claim 11 further comprising:

receiving the selected precision; and

determining a bias size, a mantissa size, and the size of the storage ~~variable~~ based on the selected precision[[;]], wherein said biasing is based on the bias size and said shifting is based on the mantissa size

~~performing said biasing based on the bias size.~~

~~performing said shifting based on the mantissa size~~.